PTO/SB/21 (12-97)
Approved for use through 9/30/00. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/879,838
Filing Date	June 12, 2001
First Named Inventor	Ralf L. BECK et al.
Group Art Unit	2122
Examiner Name	Satish RAMPURIA
Total No. of Pages in this Submission: **126**	Attorney Docket Number: ALTAWO P01AUS

ENCLOSURES *(check all that apply)*

■ Fee Transmittal Form	☐ Assignment papers *(for an Application)*	☐ After Allowance Communication to Group
■ Fee attached	☐ Drawing(s)	■ Appeal Communication to Board of Appeals and Interferences
☐ Amendment/Response	☐ Licensing-related Papers	■ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
☐ After Final	☐ Petition Routing Slip (PTO/SB/69) and Accompanying Petition	☐ Proprietary Information
☐ Affidavits/declaration(s)	☐ To Convert a Provisional Petition	☐ Status Letter
☐ Extension of Time Request	☐ Power of Attorney, Revocation Change of Correspondence Address	■ Additional Enclosure(s) *(please identify below)*:
☐ Express Abandonment Request	☐ Terminal Disclaimer	Postcard
☐ Information Disclosure Statement	☐ Small Entity Statement	Transmittal of Appeal Brief
☐ Certified Copy of Priority Document(s)	☐ Request for Refund	
☐ Response to Missing Part/s Incomplete Application		
☐ Response to Missing Parts under 37 CFR 1.52 or 1.53		

REMARKS

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual Name	Gary D. Clapp DAVIS & BUJOLD, P.L.L.C.	Reg. No. 29,055 **CUSTOMER NO. 020210**
Signature		
Date	April 8, 2005	

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 8, 2005

Type or printed name.	Gary D. Clapp	
Signature		Date: April 8, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Ralf L. BECK and Michael W. ROMANCHUK
Serial no.	:	09/879,838
Filed	:	June 12, 2001
For	:	SOFTWARE INSTRUMENTATION METHOD AND APPARATUS
Group Art Unit	:	2122
Examiner	:	Satish Rampuria
Docket	:	ALTAWO P01AUS

MAIL STOP APPEAL BRIEF - PATENTS
The Commissioner for Patents
U.S. Patent & Trademark Office
P. O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION -- 37 C.F.R. § 1.192

1. Transmitted herewith, in triplicate, is the APPEAL BRIEF in this application, with respect to the Notice of Appeal filed on January 26, 2005.

NOTE: *"Appellant must, within two months from the date of the notice of appeal under § 1.191 or within the time allowed for reply to the action from which the appeal was taken, if such time is later, file a brief in triplicate. . ." 37 C.F.R. § 1.192(a) (emphasis added).*

2. STATUS OF APPLICANT

This application is on behalf of

■ other than a small entity.

□ a small entity.

A statement:

□ is attached.

□ was already filed.

CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*
(When using Express Mail, the Express Mail label number is **mandatory**;
Express Mail certification is optional.)

I hereby certify that, on the date shown below, this correspondence is being:

MAILING

■ deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

37 C.F.R. § 1.8(a)

■ with sufficient postage as first class mail.

37 C.F.R. § 1.10*

□ as "Express Mail Post Office to Addressee"

Mailing Label No. _____ (mandatory)

Signature

Date: April 8, 2005

Gary D. CLAPP
(type or print name of person certifying)

* *Only the date of filing (§ 1.6) will be th date used in a patent term adjustment calculation, although the date on any certificate of mailing or transmission under § 1.8 continues to be taken into account in determining time lines. See § 1.703(f). Consider "Express Mail Post Office to Addressee" (§ 1.10) or facsimile transmission (§ 1.6(d) for the reply to be accorded the earliest possible filing date for patent term adjustment calculations.*

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

☐	Small entity	$250.00
■	other than small entity	$500.00

APPEAL BRIEF FEE DUE $500.00

4. EXTENSION OF TERM

NOTE: 37 C.F.R. § 1.704(b) ".....an applicant shall be deemed to have failed to engage in reasonable efforts to conclude processing or examination of an application for the cumulative total of any periods of time in excess of three months that are taken to reply to any notice or action by the Office making any rejection, objection, argument, or other request, measuring such three-month period from the date the notice of action was mailed or given to the applicant, in which case the period of adjustment set forth in § 1.703 shall be reduced by the number of days, if any, beginning on the day after the date that is three months after the date of mailing or transmission of the office communication notifying the applicant of the rejection, objection, argument, or other request and ending on the date the reply was filed. The period, or shortened statutory period, for reply that is set in the Office action or notice has no effect on the three-month period set forth in this paragraph."

NOTE: The time periods set forth in 37 C.F.R. § 1.192(a) are subject to the provision of § 1.136 for patent application. 37 C.F.R. § 1.19(d). See also Notice of November 5, 1985 (1060 O.G. 27).I

NOTE: As the two-month period set in § 1/192(a) for filing an appeal brief is not subject to the six-month maximum period specified in 35 U.S.C. § 133, the period for filing an appeal brief may be extended up to seven months. 62 Fed. Reg. 53,156; 1203 O.G. 63, at 84 (Oct. 10, 1997).

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

complete (a) or (b), as applicable)

(a) ☐ Applicant petitions for an extension of time under 37 C.F.R. § 1.136 (fees: 37 C.F.R. § 1.17(a)(1)-(5)) for th total number of months checked below:

	Extension (months)	Fee for other than Small Entity	Fee for Small Entity
☐	one month	$ 120.00	$ 60.00
☐	two months	$ 440.00	$ 225.00
☐	three months	$1,020.00	$ 510.00
☐	four months	$1,590.00	$ 795.00
☐	five months	$2,160.00	$1,080.00

FEE: **$-0-**

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

☐ An extension for _____ months has already been secured, and the fee paid therefor of $_____ is deducted from the total fee due for the total months of extension now requested.

EXTENSION FEE DUE WITH THIS REQUEST $

OR

(b) ■ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee	**$500.00**
Extension fee (if any)	$ **-0-**

TOTAL FEE DUE $500.00

6. FEE PAYMENT

■ Attached is a ■ check □ money order in the amount of **$500.00**
□ Authorization is hereby made to change the amount of $

- □ to Deposit Account No. 04-0213
- □ to Credit card as shown on the attached credit card information authorization form PTO-2038.

WARNING: Credit card information should not be included on this form as it may become public.

□ Charge any additional fees required by this paper or credit any overpayment in the manner authorized above.

A duplicate of this paper is attached.

7. FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these changes prior to action on the cases. Authorization to change the deposit account for nay fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

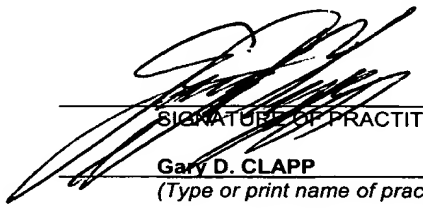
□ Of any additional extension and/or fee is required,

AND/OR

■ If any additional fee for claims is required, charge:

- ■ Deposit Account No. 04-0213
- □ Credit card as shown on the attached credit card information authorization form PTO-2038.

WARNING: Credit card information should not be included on this form as it may become public.



SIGNATURE OF PRACTITIONER

Gary D. CLAPP
(Type or print name of practitioner)

DATE: **April 8, 2005**

REG. NO.: **29,055**

Customer No. **020210**

DAVIS & BUJOLD, P.L.L.C.
FOURTH FLOOR
500 NORTH COMMERCIAL STREET
MANCHESTER, NH 03101-1151

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF APPEALS AND INTERFERENCES

****

Appeal Number

In re Application of	:	Ralf L. BECK and Michael W. ROMANCHUK
Serial no.	:	09/879,838
Filed	:	June 12, 2001
For	:	SOFTWARE INSTRUMENTATION METHOD AND APPARATUS
Group Art Unit	:	2122
Examiner	:	Satish RAMPURIA
Docket	:	ALTAWO P01AUS

****

APPELLANT'S BRIEF

4/08/05 PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Ralf L. BECK and Michael W. ROMANCHUK
Serial no.	:	09/879,838
Filed	:	June 12, 2001
For	:	SOFTWARE INSTRUMENTATION METHOD AND APPARATUS
Group Art Unit	:	2122
Examiner	:	Satish RAMPURIA
Docket	:	ALTAWO P01AUS

MAIL STOP APPEAL BRIEF - PATENTS
The Hon Commissioner of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

This Appeal Brief is being filed in support of Appellant's Notice of Appeal mailed on March 23, 2005 because of the final rejection, mailed January 26, 2005, of claims 1-14 issued by the Primary Examiner.

I. REAL PARTY IN INTEREST: The real party in interest is:

OPNET Technologies Inc.

II. RELATED APPEALS AND INTERFERENCES: There are no related appeals or interferences in respect of the instant or any related patent application.

III. STATUS OF CLAIMS:

A. Total Number Of Claims In Application: 14

Claims in the Application are: 1 through 14

B. Status Of All The Claims:

1. Claims Canceled: None

2. Claims Withdrawn From Consideration But Not Canceled: None

3. Claims Pending: 1 through 14

4. Claims Allowed: None

5. Claims Rejected: 1 through 14

C. Claims On Appeal: 1 through 14

IV. STATUS OF AMENDMENTS

On October 5, 2004 and in reply to an Official Action mailed on July 6, 2004, the Applicant filed amendments of claims 1-5 and 7-14, including amendments addressing rejections of the claims under 35 U.S.C. § 112, amendments addressing certain informalities in the specification, and arguments distinguishing the amended claims over the prior art cited in rejection of the claims in the Official Action of July 6, 2004. The amendments and arguments submitted on July 6, 2002 were entered into the record of this case prior to issuance of a Final Action mailed January 26, 2005 in which the Examiner essentially repeated the rejections over the prior art stated in the Official Action of July 6, 2004. No further amendment has been filed subsequent to the response of October 5, 2004.

V. SUMMARY OF THE INVENTION

The present invention relates to improvements in the instrumentation of computer program code, that is, software and, in particular, to a method and mechanism for instrumenting object oriented virtual-machine-executable software comprised of class files containing bytecode wherein each class file is defined by a class–associated with a class-hierarchy location and has a corresponding class name and wherein the instrumentation is implemented without modification of the original bytecode of the class files.

The instrumentation processes create a completely new instrumented class 140x, 140y, 140z, 902x, 1002y, 1102z that is inserted into an execution path of the target class 110a, 110b, 110c, 140, 502, 1102C such that a virtual machine 120 will process the newly created instrumented class 140x, 140y, 140z, 902x, 1002y, 1102z, thereby providing the desired instrumentation, while the bytecode of the original target class 110a, 110b, 110c, 140, 502, 1102C remains unmodified.

According to the present invention, a class identifier identifies a class 110, 140 included in the software as a target class 110, 140 to be instrumented wherein the target class 110, 140 is associated with a first class-hierarchy location and with a first class. The target class 110, 140 is then implemented by an instrumentation process that is selected from a set of three instrumentation processes that in each instance generates a new instrumented class to represent the selected target class and without modifying the bytecode of the selected target class. The specific process that is to be used in a given instance is dependent upon the circumstances and the type of instrumentation to provided. In particular, and expressed very briefly and generally, one instrumentation process instruments invocations of the target class by other classes, a second instrumentation process instruments invocations of other classes, and a third instrumentation process instruments the target class itself.

The first instrumentation process is defined and implemented by the steps of (a) creating a new instrumented class 140x, 902x separate from the target class 110b,140b, 502b, (b) adding instrumentation to the new instrumented class 140x, 902x, and (c) assigning the new instrumented class 140x, 902x to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class 110b, 140b, 502b. The instrumentation thereby comprised of additions to the bytecode of the newly created class 140x, 902x rather than as modifications to the bytecode of the original target class 110b, 140b, 502b. The bytecode of the target class 110b, 140b, 502b thereby remains in its original, unmodified form, but is instrumented by means of the instrumentation added to the newly created class 140x, 902x that is effectively located in the execution path(s) leading to the target class 110b, 140b, 502b. The new instrumented class 140x, 902x is, therefore, not a modified version or the original target class 110b, 140b, 502b, but is an entirely new class 140x, 902x created independently and separately from the target

class 110b, 140b, 502b for the specific purpose of providing instrumentation for the target class 110b,140b, 502b.

The second instrumentation process is defined and implemented by the steps of (a) creating a new instrumented class 140y,1002y separate from the target class 110b, 140b, 502b, (b) adding instrumentation to the new instrumented class 140y, 1002y, (c) assigning the new instrumented class 140y, 1002y to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class 110b, 140b, 502b, (d) assigning the first class name to the new instrumented class 140y, 1002y and (e) assigning a second class name to the target class 110b, 140b, 502b. The instrumentation is again comprised of additions to the bytecode of the newly created class 140y, 1002y rather than as modifications to the bytecode of the original target class 110b, 140b, 502b, so that the bytecode of the target class 110b, 140b, 502b again remains in its original, unmodified form but is instrumented by means of the instrumentation added to the newly created class 140y, 1002y that is effectively located in the execution path(s) leading from the target class 110b, 140b, 502b. Again, the new instrumented class 140y, 1002y is not a modified version or the original target class 110b, 140b, 502b but is an entirely new and separate class created independently and separately from the target class 110b, 140b, 502b for the specific purpose of providing instrumentation for the target class 110b, 140b, 502b.

The third instrumentation process is defined and implemented by the steps of (a) creating a new instrumented class 140z, 1102z separate from the target class 110c, 1102c and including instrumentation, (b) assigning the first class name to the new instrumented class 140z, 1102z and (c) assigning a second class name to the target class 110c, 1102c whereby the new instrumented class 140z, 1102z is recognized as the target class 110c, 1102c and the target class 110c, 1102c becomes inactive because it will be unrecognized by a virtual machine 120 executing the software. The instrumentation is again located in the bytecode of the newly created instrumented

class 140z, 1102z rather than being implemented as modifications to the bytecode of the original target class 110c, 1102c, so that the bytecode of the target class 110c, 1102c again remains in its original, unmodified form. The original target class 110c, 1102c is thereby instrumented by means of the instrumentation in the newly created class 140z, 1102z that effectively replaces the target class 110c, 1102c by being assigned the first class name that was originally assigned to the target class 110c, 1102c. At the same time, the target class 110c, 1102c becomes unrecognized and thereby unexecuted by being assigned the second class name, which will not be recognized by the virtual machine 120. Again, the new instrumented class 140z, 1102z is not a modified version or the original target class 110c, 1102c, but is an entirely new separate class 140z, 1102z created independently and separately from the target class 110c, 1102c for the specific purpose of providing instrumentation for the target class 110c, 1102c.

VI. ISSUES:

The issues presented for appeal are:

(a) Whether claims 1, 8, 9, 10, 11, 12 and 13 are unpatentable as obvious under 35 U.S.C. § 103(a) over U.S. Patent No. 6,314,558 to Angel et al. for BYTE CODE INSTRUMENTATION, hereafter referred to as "Angel et al. `558", in view of U.S. Patent No. 6,026,237 to Berry et al. for SYSTEM AND METHOD FOR DYNAMIC MODIFICATION OF CLASS FILES, hereafter referred to as "Berry et al. `237"; and,

(b) Whether claims 2, 3, 4, 5, 6, 7 and 14 are unpatentable as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of U.S. Patent No. 6,405,367 to Bryant et al. for APPARATUS AND METHOD FOR INCREASING THE PERFORMANCE OF JAVA PROGRAMS RUNNING ON A SERVER, hereafter referred to as "Bryant et al. `367".

VII. GROUPING OF CLAIMS:

For purposes of the present Appeal, the grouping of claims as defined by the claim rejections are:

Group I: claims 1, 8, 9 and 13 under grounds for rejection (a) wherein claims 8, 9 and 13 are believed to stand or fall with the allowability of claim 1;

Group II: claim 10 under grounds for rejection (a);

Group III: claim 11 under grounds for rejection (a);

Group IV: claim 12 under grounds for rejection (a);

Group V: claims 2, 3, 5, 6, 7 and 14 under groups for rejection (b) wherein dependent claims 2, 3, 5, 6, 7 and 14 are believed to stand or fall with the allowability of claim 1; and

Group VI: claim 4 under grounds for rejection (b).

VIII. ARGUMENT:

(A) Group I - Claims 1, 8, 9, and 13 under grounds for rejection (a):

Claims 1 and 13 are equivalent and parallel independent claims wherein claim 1 is a method claim and claim 13 is a corresponding apparatus claim and wherein claims 8 and 9 are method claims dependent from claim 1, this group of claims is thereby discussed with respect to claim 1.

It is the Applicants' belief that the rejection of claims 1, 8, 9 and 13 as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237 is in error for the following reasons.

First considering Angel et al. `558, it has been described above and is recited in claim 1, and in claim 13, that the method and apparatus of the present invention instruments a target class by (a) creating a new instrumented class that is separate and independent from the target class rather than by modifying the existing target class and (b) inserting the new instrumented class in the path into or out of the target class or in replacement for the target class while the target class

remains unmodified. The present invention thereby instruments a target class by creating a completely new instrumented class that is inserted into an execution path of the target class such that a virtual machine will process the newly created instrumented class, thereby providing the desired instrumentation while the bytecode of the original target class remains unmodified.

In basic contrast from the present invention, Angel et al. `558 clearly teaches the instrumentation of a class by modifying the bytecode of the class to be instrumented, so that Angel et al. `558 thereby (a) does not create a new instrumented class that is separate and independent from the class to be instrumented and (b) clearly teaches and requires the modification of the bytecode of the class to be modified. These teachings of Angel et al. `558, and the resulting distinctions from the present invention, are clearly and completely described in Angel et al. `558 in, for example, the Abstract, and at column 3, lines 16-33, of the Specification. These teachings of Angel et al. `558 are further fully supported and confirmed in the Detailed Description of Angel et al. `558 appearing, for example, in column 20, line 44, through column 32, line 50.

Further in this regard, the Examiner has held that at column 3, lines 22-23 of Angel et al. `558 teaches "adding instrumentation to the target class without modifying bytecode within the target class" (column 3, lines 22-23).

The Applicant respectfully disagrees with the Examiner on the grounds that this statement from Angel et al. `558 has been taken out of context in such a manner as to severely change the original meaning of the statement. Angel et al. `558, in fact, clearly describes that the instrumentation is accomplished by modifying the bytecode of the target class. Column 3, lines 22-23 of the Angel et al. `558 specification refers to "[i]nstrumenting a portion of the byte code corresponding to a method call", which, placed in context, clearly states that the target class is instrumented by instrumenting a portion of the byte code of the target class, that is, by changing

that portion of the original byte code of the target class. In support of this position, it must be noted that other portions of Angel et al. `558 also explicitly state that the instrumentation of a class is accomplished by modifying the byte code of the class.

The Examiner also refers column 20, lines 26-62, of Angel et al. `558 as describing "causing a virtual machine to process as the target class the class assigned the first class name" and describing that "Byte code may be instrumented by instrumenting each class as the class is loaded by the VM runtime system."

The Applicant respectfully disagrees with the Examiner with regard to this interpretation of Angel et al. `558 because the two statements refer to entirely different operations and because neither describes the essential aspect of the present invention wherein the instrumentation of a target class is accomplished without modification of the original byte code of the target class.

The first statement refers to causing a virtual machine to process as the target class a class that is assigned the first class name, which is the method used by the present invention to cause the virtual machine to process the instrumented class in place of the target class when the virtual machine is processing the software but which is not the essential aspect of the present invention.

The second statement, in turn, merely states that the byte code of a class file may be instrumented when the class is loaded for execution, which is an entirely different operation from that of the first statement. The second statement, however, does not describe at all how the byte code of a class file is to be instrumented, but only when it may be instrumented, and thereby does not relate to the present invention at all. In this regard, it must again be noted that the other teachings in Angle et al. `558 teach directly away from the present invention by explicitly teaching that the byte code is to be instrumented by modifying the original byte code of the target class. As a consequence, the instrumentation operation referred to in the second statement can only refer

to instrumenting the byte code by modifying the original byte code of the target class, which is diametrically opposed to the present invention.

It is, therefore, the position of the Applicant that, for the reasons discussed above, Angel et al. `558 does not teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claim 1, or in claim 13, and that Angel et al. `558, in fact, teaches directly away from the present invention.

Angel et al. `558 therefore does not teach or suggest the present invention as recited in claim 1 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 does not teach or suggest the recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form."

Next, considering the teachings of Berry et al. `237, this reference is similar to Angel et al. `558, and thereby again in basic contrast from the present invention, in clearly teaching the instrumentation of a class by modifying the bytecode of the class to be instrumented. Berry et al. `237 thereby does not create a new instrumented class that is separate and independent from the class to be instrumented and clearly teaches and requires the modification of the bytecode of the class to be modified. Such teaching of Berry et al. `237, and the resulting distinctions from the present invention, are clearly and completely described in Berry et al. `237 in, for example, the Abstract, in Fig. 2 and at column 3, line 57 through column 4, line 27 of the Specification.

These teachings of Berry et al. `237 are further fully supported and confirmed in the more detailed description of Berry et al. `237 appearing, for example, in columns 7 through 9. For example, at column 5, lines 20-24, Berry et al. `237 states that "[c]ode may be added to, deleted from, or modified in the Java class file for many reasons, including instrumentation, benchmarking, performance tuning, modifying functionality, applying functional or performance patches". It is the Applicants' position that Berry et al. `237 thereby does, in fact, teach the modification of an existing file that is to be instrumented, rather than the creation of a new file containing the instrumentation.

In this regard, it must also be noted that the Applicant respectfully disagrees with the Examiner's interpretation of the teachings of Berry et al. `237. For example, the Examiner cites Berry et al. `237 at column 7, line 15, wherein Berry et al. `237 states "[t]he modified class file is then created (step 522)", and at column 5, line 21, wherein Berry et al. `237 states "adding code for the purposes of instrumentation". The Examiner interprets these statements as saying that Berry et al. `237 creates a separate new class and adds instrumentation to the separate new class,

thereby effectively holding that the modification of an existing class file is the same as the creation of a separate file.

The Applicant respectfully disagrees with the Examiner's interpretations of Berry et al. `237 and it is the position of the Applicant that these statements by Berry et al. `237 are more properly interpreted as stating that Berry et al. `237 modifies the original class file to create an instrumented class file, rather than as stating that Berry et al. `237 creates an entirely new file that is separate and distinct from the original class file, as is done in the present invention. It must be further noted that the interpretation that Berry et al. `237 modifies an existing class file to create an instrumented class rather than creating a new instrumented class file that is separate and distinct from the original class file is in full agreement with the remaining other teachings in Berry et al. `237.

It is, therefore, the position of the Applicant that, for the reasons discussed above, Berry et al. `237 does not teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claim 1, or in claim 13, and that Berry et al. `237, in fact, teaches directly away from the present invention.

Berry et al. `237, therefore, does not teach or suggest the present invention as recited in claim 1 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Berry et al. `237 does not teach or suggest the recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class,

assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form."

Therefore considering the teachings and suggestions of Angel et al. `558 in view of Berry et al. `237, it is clear that both Angel et al. `558 and Berry et al. `237 clearly teach the instrumentation of a class by modifying the bytecode of the class to be instrumented, so that both Angel et al. `558 and Berry et al. `237 clearly teach and require the modification of the bytecode of the class to be modified. Neither Angel et al. `558 nor Berry et al. `237 teaches or suggest the creation of a new instrumented class that is separate and independent from the class to be instrumented, so that no combination of Angel et al. `558 nor Berry et al. `237 does or could teach or suggest the present invention under the requirements and provisions of 35 U.S.C. § 103.

For these reasons, therefore, it is the Applicant's position that the combination of Angel et al. `558 in view of Berry et al. `237 does not teach, suggest or hint at the present invention that are recited in claims 1 and 13 and that Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237, in fact, teaches directly away from the present invention.

Angel et al. `558 in view of Berry et al. `237, therefore, does not teach or suggest the present invention as recited in claim 1 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. That is, Angel et al. `558 in view of Berry et al. `237 does not teach or suggest the creation a new instrumented class that is separate and independent from the

target class and the insertion of the new instrumented class into the software in such as way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class.

In particular, Angel et al. `558 in view of Berry et al. `237 does not teach or suggest the recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form"

to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

(B) Group II - Claim 10 under grounds for rejection (a):

Claim 10 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1, but is treated herein as standing or falling independently of claim 1 as

claim 1 is a genus claim reciting the method of the present invention while claim 10 is a species claim directed to the method of the present invention as embodied for a specific circumstance.

It is the Applicants' belief that the rejection of claim 10 as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237 is in error for the following reasons.

In the rejection of claim 10, the Examiner has stated that Angel et al. `558 does not teach virtual machine that is a Java virtual machine and the addition of instrumentation to or modification of the target class by the modification of memory locations as recited in claim 10, but that Berry et al. `237 describes a Java virtual machine and such modifications to a class file.

In response, it must be noted that the present invention is not directed to these methods for modifying class files per se, but to a method for instrumenting a target class file by creating a new instrumentation file that is separate and independent from the target class and placing the instrumentation in the new instrumentation file while leaving the original target class unmodified, as recited in claim 1 and as thereby recited in claim 10. The method for class file modification recited in claim 10 is used to effectively redirect the virtual machine away from the unmodified target class and to the new created instrumented class. That is, these methods are employed, according to the present invention, to cause the newly created instrumented class to appear to the virtual machine as the original target class while causing the virtual machine to not recognize the original target class as the original target class.

In contrast from the method recited in claim 10, the methods taught by Berry et al. `237 are employed by Berry et al. `237 to modify the "target" class to thereby directly instrument the target class files, rather than, as in the present invention, to cause the newly created instrumented class to appear to the virtual machine as the original target class while causing the virtual machine to not recognize the original target class as the original target class. Therefore, while the class file modification methods taught by Berry et al. `237 are somewhat analogous to those recited in

claim 10 to redirect the virtual machine, the use and purpose of the methods of the present invention are fundamentally distinguished from the uses and purposes of those methods by Berry et al. `237.

As such, it is the position of the Applicant that although the specific steps recited in claim 10 are similar to those employed in the class file modification methods used by Berry et al. `237, the method of the present invention as recited in claim 10, which includes the method steps recited in claim 1, are fully distinguished from the class file modification methods for Java virtual machines as taught by Berry et al. `237 by being directed to fundamentally different purposes and results.

In addition, and further fundamental distinction between the present invention and the teachings of Berry et al. `237, it must be noted that claims 8, 9 and 10 are dependent from claim 1 and thereby incorporate be dependence all recitations and limitations of claim 1. For the reasons discussed herein above, claim 1 is fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claim 10 by dependence therefrom, claim 10 is thereby also fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103.

Angel et al. `558 in view of Berry et al. `237, therefore, does not teach or suggest the present invention as recited in claim 10 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 in view of Berry et al. `237 does not teach or suggest the recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a

class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form"; and,

(Iv) the step of "adding instrumentation to the target class including adding, removing, or reordering or substituting a memory location representing the target class or changing the contents of a memory l location representing the target class"

to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

(C) Group III - Claim 11 under grounds for rejection (a):

Claim 11 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1, but is treated herein as standing or falling independently of claim 1 as claim 1 is a genus claim reciting the method of the present invention while claim 11 is a species claim directed to the method of the present invention as embodied for a specific circumstance.

It is the Applicants' belief that the rejection of claim 11 as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237 is in error for the following reasons.

In the rejection of claim 11 under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237, the Examiner states that Angel et al. `558 does not have teachings relevant to the recitations of claim 11 but is instead relevant to the recitations of parent claim 1 from which claim 11 is dependent, and cites Berry et al. `237 at column 1, lines 60-62 as teaching superclasses as recited in claims 11.

The Applicant has discussed the relevance of Angel et al. `558 to the recitations of parent claim 1 herein above and will not again repeat those discussions in detail herein. In summary, however, and as discussed above, the present invention is directed to a method for instrumenting byte code by creating a new instrumented class that is separate and independent from the target class to be instrumented and causing the system to executed the newly created class rather than the target class while leaving the target class in its original form. In complete contrast, Angel et al. `558 teaches the instrumentation of a class by modifying the original bytecode of the class to be instrumented.

With regard to the teachings of Berry et al. `237 with respect to claim 11, the Applicant respectfully disagrees with the Examiner's interpretation of the relationship of the teachings of Berry et al. `237 regarding superclasses and the recitations of claim 11. The cited statement in Berry et al. `237 is in the Background of the Invention and is part of a generalized description of the well known and commonly used object oriented Java class files and their hierarchical relationship to one another, such as the relationship of a class to a "superclass" of which it is a member. As such, the description of Java class files, classes and superclasses is no more than background information and has little relevance to various actual uses of such class/superclass relationships in the present invention.

For example, the descriptions of Java file classes and superclasses in Berry et al. `237 does not describe or even suggest the use, as in the present invention as recited in claim 11, of class

and superclass relationships to insert a new instrumented class in the execution/invocation path to or from a target class to thereby instrument the unmodified target class by means of the new instrumented class. Berry et al. `237 instead describes only the well known and long established organizational and structural arrangements for object oriented code, rather than describing the use of such organizational and structural restrictions for new purposes not set forth in the original, basic structural restrictions imposed in object oriented programming for other reasons.

As such, it is the position of the Applicant that the cited teachings of Berry et al. `237, while pertinent to Java class files, classes and superclasses, is not relevant to the specific aspects of the present invention that are recited in claim 11.

In addition, and in fundamental distinction between the present invention and the teachings of Berry et al. `237 and of Angel et al. `558 in view of Berry et al. `237, it must be noted that claim 11 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1.

For the reasons discussed herein above, therefore, claim 1 is fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claim 11 by dependence therefrom, claim 11 is thereby also fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103.

Angel et al. `558 in view of Berry et al. `237, therefore, does not teach or suggest the present invention as recited in claim 11 to those of ordinary skill in the arts under the requirements

and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 in view of Berry et al. `237 does not teach or suggest the incorporated recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form";

or the recitations of claim 11 of

"the step of assigning the new instrumented class to a class hierarchy location adjacent to and above the class hierarchy location associated with the target class includes modifying the new instrumented class to recognize a super class associated with the target class as the super class associated with the new instrumented class" and

"modifying the target class to recognize the new instrumented class as the super class associated with the target class."

to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

(D) Group IV - Claim 12 under grounds for rejection (a):

Claim 12 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1, but is treated herein as standing or falling independently of claim 1 as claim 1 is a genus claim reciting the method of the present invention while claim 12 is a species claim directed to the method of the present invention as embodied for a specific circumstance.

It is the Applicants' belief that the rejection of claim 12 as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237 is in error for the following reasons.

In the rejection of claim 12 under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Berry et al. `237, the Examiner states that Angel et al. `558 does not have teachings relevant to the recitations of claim 12 but is instead relevant to the recitations of parent claim 1 from which claim 12 is dependent, and cites Berry et al. `237 at column 1, lines 60-62 as teaching superclasses as recited in claims 12.

The Applicant has discussed the relevance of Angel et al. `558 to the recitations of parent claim 1 herein above and will not again repeat those discussions in detail herein. In summary, however, and as discussed above, the present invention is directed to a method for instrumenting byte code by creating a new instrumented class that is separate and independent from the target class to be instrumented and causing the system to executed the newly created class rather than the target class while leaving the target class in its original form. In complete contrast, Angel et al. `558 teaches the instrumentation of a class by modifying the original bytecode of the class to be instrumented.

With regard to the teachings of Berry et al. `237 with respect to claim 12, the Applicant respectfully disagrees with the Examiner's interpretation of the relationship of the teachings of Berry et al. `237 regarding superclasses and the recitations of claim 12. The cited statement in Berry et al. `237 is in the Background of the Invention and is part of a generalized description of

the well known and commonly used object oriented Java class files and their hierarchical relationship to one another, such as the relationship of a class to a "superclass" of which it is a member. As such, the description of Java class files, classes and superclasses is no more than background information and has little relevance to various actual uses of such class/superclass relationships in the present invention.

For example, the descriptions of Java file classes and superclasses in Berry et al. `237 does not describe or even suggest the use, as in the present invention as recited in claim 12, of class and superclass relationships to insert a new instrumented class in the execution/invocation path to or from a target class to thereby instrument the unmodified target class by means of the new instrumented class. Berry et al. `237 instead describes only the well known and long established organizational and structural arrangements for object oriented code, rather than describing the use of such organizational and structural restrictions for new purposes not set forth in the original, basic structural restrictions imposed in object oriented programming for other reasons.

As such, it is the position of the Applicant that the cited teachings of Berry et al. `237, while pertinent to Java class files, classes and superclasses, is not relevant to the specific aspects of the present invention that are recited in claim 12.

In addition, and in fundamental distinction between the present invention and the teachings of Berry et al. `237 and of Angel et al. `558 in view of Berry et al. `237, it must be noted that claim 12 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1.

For the reasons discussed herein above, therefore claim 1 is fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are

incorporated into claim 12 by dependence therefrom, claim 12 is thereby also fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and the combination of Angel et al. `558 in view of Berry et al. `237 under the requirements and provisions of 35 U.S.C. § 103.

Angel et al. `558 in view of Berry et al. `237, therefore, does not teach or suggest the present invention as recited in claim 12 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 in view of Berry et al. `237 does not teach or suggest the incorporated recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form"; or

the recitations of claim 12 of:

"the step of assigning the new instrumented class to a class hierarchy location adjacent to and below the class hierarchy location associated with the target class includes modifying the new instrumented class to recognize the target class; as the super class associated with the new instrumented class"

to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

(E) Group V - Claims 2, 3, 5, 6, 7 and 14 under groups for rejection (b):

Claims 2, 3, 5 and 6 are directly dependent from claim 1 and thereby incorporate all limitations and recitations of claim 1 while claim 7 is dependent from claim 6 and thus incorporates all recitations and limitations of claims 1 and 6. Claim 14 is dependent from claim 13, which is an apparatus claim that is generally parallel and equivalent to claim 1, and which thereby incorporates all recitations and limitations of claim 13 and thereby effectively incorporates the essentially the same limitations and recitations as are found in claim 1.

It is the Applicants' belief that the rejection of claims 2, 3, 5, 6, 7 and 14 as unpatentable as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Bryant et al. `367 is in error for the following reasons.

The Examiner cites the teachings of Angel et al. `558 with respect to the rejections of claims 2, 3, 5, 6, 7 and 14 with regard to the recitations of claim 1 that are incorporated therein by dependency, but also states with regard to claims 2, 3, 5 and 6 that Angel et al. `558 does not teach the operations of suspending the operation of a virtual machine between the loading and linking of a target class. The Examiner further states, however, that Bryant et al. `367 does contain such teachings with respect t claims 2, 3, 5 and 6 and cites Bryant et al. `367 at column 6, lines 56-57 and 57-60.

Again, the Applicant respectfully disagrees with the Examiner's interpretation of the teachings of Angel et al. `558 with regard to the recitations of claim 1 that are incorporated into

claims 2, 3, 5, 6 7 and 14 on the grounds that these teachings from Angel et al. `558 represent only a small part of the teachings of the Angel et al. `558 specification. It is the Applicants' position that the actual meaning of a prior art reference can be fully understood only when the entire reference and all parts thereof are taken in context and that the cited sections from Angel et al. `558 have been taken out of context in such a manner as to severely change the original meaning of the statements.

Angel et al. `558 in fact clearly describes that the instrumentation of the bytecode is to be accomplished by modifying the bytecode of the target class to include the necessary instrumentation. For example, column 3, lines 22-23 of the Angel et al. `558 specification refers to "[i]nstrumenting a portion of the byte code corresponding to a method call", which, placed in context, clearly states that the target class bytecode is instrumented by changing that portion of the original byte code of the target class. Other portions of Angel et al. `558 also explicitly state that the instrumentation of a class is accomplished by modifying the byte code of the class.

In contrast, the method of the present invention instruments the target class byte code by creating a new instrumented class that is separate and independent from the target class and causing the system to executed the newly created class rather than the target class, thereby leaving the target class in its original, unmodified form. As discussed above, therefore, it is the Applicants' position that Angel et al. `558 does not teach or suggest the present invention as recited in parent claim 1, or as recited in claims 2, 3, 5, 6, 7 or 14 as incorporated from claim 1.

Therefore considering the teachings of Bryant et al. `367, Bryant et al. `367 teaches a method for increasing the performance of the execution of a Java application on a Java server by using Java application code and functions of the application on the server as a library of classes and methods. When the application is to be executed, an object file calls the appropriate Java server process of the application on the Java server and the Java server process forks itself as a

child server from the already existing parent server in the conventional manner and causes the child server to run the already loaded classes and methods of the application code. That is, the object calling the parent Java server communicates with the parent Java server while, in actuality the parent Java server has forked the child Java server, which is executing the methods and classes of the parent Java server and exchanging calls and returns, including data, between the child and parent Java servers. It therefore appears to the calling object that the parent Java server is executing the methods and classes of the called process while, in actuality, it is the child server that is executing the methods and classes.

As a consequence, and as specifically taught by Bryant et al. `367, it is only necessary to load the classes and methods once, that is, in the original parent server when the Java virtual machine is started. According to Bryant et al. `367, this method enhances the speed of execution of a Java process because it is faster to fork an already running Java server to a child server in order to execute the necessary classes and methods than it is to load the Java virtual machine, that is, the parent Java server, at each execution of the Java process.

A complete reading of Bryant et al. `367, therefore, shows that the cited portions of the Bryant et al. `367 specification in fact describe the interaction between the parent Java server and the child Java server and does not relate or pertain to the structural or operational relationships between class files within a Java process executing on a single virtual machine, or server. As such, it is apparent that the teachings of Bryant et al. `367 are, in fact, not material or relevant to the present invention as recited in either the recitations of claim 1, or in the recitations of claims 2, 3, 5, 6, 7 or 14.

In addition, none of Bryant et al. `367, Angel et al. `558 or Berry et al. `237 teach, or suggest fundamental and essential aspects of the present invention that are recited in claim 1, that is, and in particular, instrumenting a target class byte code by creating a new instrumented

class that is separate and independent from the target class and causing the system to executed the newly created class rather than the target class, thereby leaving the target class in its original, unmodified form. Angel et al. `558 and Berry et al. `237, in fact, teach directly away from the present invention by explicitly teaching the direct opposite of the present invention, that is, by teaching modification of the original byte code of the target class in order to instrument the target class. Bryant et al. `367, in turn, is not material or pertinent to the present invention as recited in claim 1 because Bryant et al. `367 does not address this subject matter or contain any teachings at all pertaining to this subject matter.

For the reasons discussed herein above, therefore, claim 1 is fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and Bryant et al. `367 and over and from all combinations of Angel et al. `558 in view of Bryant et al. `367 or Angel et al. `558 in view of Berry et al. `237 or Angel et al. `558 in view of Berry et al. `237 in view of Bryant et al. `367 under the requirements and provisions of 35 U.S.C. § 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claims 2, 3, 5, 6, 7 and 14 by direct or indirect dependence therefrom, claims 2, 3, 5, 6, 7 and 14 are thereby also fully and patentably distinguished over and from Angel et al. `558 and Berry et al. `237 and Bryant et al. `367 and all combinations of Angel et al. `558 in view of Berry et al. `237 in view of and Bryant et al. `367 under the requirements and provisions of 35 U.S.C. § 103.

Angel et al. `558 in view of Bryant et al. `367, therefore, does not teach or suggest the present invention as recited in claims 2, 3, 5, 6, 7 or 14 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the incorporated recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a

class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the recitations of claim 2 of:

"(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the target class;

(ca) after completing step (c), un-suspending operation of the virtual machine."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the recitations of claim 3 of:

"(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the target class to execute a software program that executes step (c);

(ca) upon termination of the software program that executes step (c) resuming the operation of the virtual machine."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the recitations of claim 5 of:

"wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using an interface provided by the virtual machine."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the recitations of claim 6 of:

"wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using a virtual machine that is modified to perform this capability."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the recitations of claim 14 of:

"further including class identification apparatus for identifying a class included within object oriented virtual-machine-executable software, the class identification apparatus comprising:

a class query engine which receives as input a set of class attribute names and associated value descriptions; and

a class searcher that sequentially searches classes included in the software and performs a predetermined action upon identifying a class that possesses a set of class attribute names and associated values that match the set of class attribute names and associated value descriptions as the target class"

to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

(F) Group VI - Claim 4 under grounds for rejection (b):

Claim 4 is dependent from claim 1 and thereby incorporates by dependence all recitations and limitations of claim 1, but is treated herein as standing or falling independently of claim 1 as claim 1 is a genus claim reciting the method of the present invention while claim 4 is a species claim directed to the method of the present invention as embodied for a specific circumstance.

It is the Applicants belief that the rejection of claim 4 as obvious under 35 U.S.C. § 103(a) over Angel et al. `558 in view of Bryant et al. `367 is in error for the following reasons.

The Examiner cites the teachings of Angel et al. `558 with respect to the rejections of claim 4 with regard to the recitations of claim 1 that are incorporated therein by dependency, but also states with regard to claim 4 that Angel et al. `558 does not teach the recitations and limitations of claim 4. The Examiner cites Bryant et al. `367, however, and in particular at column 26, lines 21-16, at the remainders of columns 26 and 27, and at column 7, lines 45-53, as teaching the elements of claim 4 that are not taught by Angel et al. `558.

Again, the Applicant respectfully disagrees with the Examiner's interpretation of the teachings of Angel et al. `558 with regard to the recitations of claim 1 that are incorporated into claim 4 on the grounds that these teachings from Angel et al. `558 represent only a small part of the teachings of the Angel et al. `558 specification. It is the Applicants' position that the actual meaning of a prior art reference can be fully understood only when the entire reference and all parts thereof are taken in context and that the cited sections from Angel et al. `558 have been taken out of context in such a manner as to severely change the original meaning of the statements.

Angel et al. `558 in fact clearly describes that the instrumentation of the bytecode is to be accomplished by modifying the bytecode of the target class to include the necessary instrumentation. For example, column 3, lines 22-23 of the Angel et al. `558 specification refers

to "[i]nstrumenting a portion of the byte code corresponding to a method call", which, placed in context, clearly states that the target class bytecode is instrumented by changing that portion of the original byte code of the target class. Other portions of Angel et al. `558 also explicitly state that the instrumentation of a class is accomplished by modifying the byte code of the class.

In contrast, the method of the present invention instruments the target class byte code by creating a new instrumented class that is separate and independent from the target class and causing the system to executed the newly created class rather than the target class, thereby leaving the target class in its original, unmodified form. Angel et al. `558 thereby and, in fact, teaches directly away from the present invention by explicitly teaching the direct opposite of the present invention, that is, by teaching modification of the original byte code of the target class in order to instrument the target class.

Next considering the teachings of Bryant et al. `367, the Applicant again respectfully disagrees with the Examiner's interpretation of the teachings of Bryant et al. `367 on the grounds that the cited teachings are extracted from the context of the Bryant et al. `367 specification and do not consider the entirety of the Bryant et al. `367 teachings.

In particular, and as discussed above, Bryant et al. `367 teaches a method for increasing the performance of the execution of a Java application on a Java server by using Java application code and functions of the application on the server as a library of classes and methods. When the application is to be executed, an object file calls the appropriate Java server process of the application on the Java server and the Java server process forks itself as a child server from the already existing parent server in the conventional manner and causes the child server to run the already loaded classes and methods of the application code. That is, the object calling the parent Java server communicates with the parent Java server while, in actuality the parent Java server has forked the child Java server, which is executing the methods and classes of the parent Java

server and exchanging calls and returns, including data, between the child and parent Java servers. It, therefore, appears to the calling object that the parent Java server is executing the methods and classes of the called process while, in actuality, it is the child server that is executing the methods and classes.

A complete reading of Bryant et al. `367 shows that the cited portions of the Bryant et al. `367 specification refer to the method by which the child server identifies and locates the required classes and methods in the parent server, which as described is employed as a library of classes and methods used in the child server process, which as described is executed in place of the server process residing in the parent server. State another way, Bryant et al. `367 describes a method by which one server, that is, the child server, can identify and locate a plurality of classes and methods residing in a second server, that is, the parent server.

The operation described by Bryant et al. `36 is, therefore, fundamentally and specifically distinguished from the operations of the present invention as recited in claim 4, which pertains to a method for identifying a specific single target class in a Java virtual machine by searching for and matching a specified set of attribute names and associated values within a single Java virtual machine and a single process. In fundamental contrast from the present invention as recited in claim 4, the method taught by Bryant et al. `367 pertains to the identification and location by one process running on one virtual machine of methods and classes of a library of methods and classes residing in another process in another virtual machine and thereby has no relevance or materiality to the present invention as recited in either of claims 1 or 4.

For the reasons discussed herein above, therefore, claims 1 and 4 are each fully and patentably distinguished over and from Angel et al. `558 and Bryant et al. `367 and over and from all combinations of Angel et al. `558 in view of Bryant et al. `367 under the requirements and provisions of 35 U.S.C. § 103. For the same reasons, and because the recitations and limitations

of claim 1 are incorporated into claim 4 by dependence therefrom, claim 4 is thereby further fully and patentably distinguished over and from Angel et al. `558 and Bryant et al. `367 and all combinations of Angel et al. `558 in view of Bryant et al. `367 under the requirements and provisions of 35 U.S.C. § 103.

Angel et al. `558 in view of Bryant et al. `367, therefore, does not teach or suggest the present invention as recited in claims 1 or 4 to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103. In particular, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the incorporated recitations of claim 1 of:

"(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;" or

"(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form;" or

"(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form."

In a like manner, Angel et al. `558 in view of Bryant et al. `367 does not teach or suggest the incorporated recitations of claim 4 of:

"the step of identifying the target class included within the virtual machine executable software includes the steps of:

(a) specifying a set of class attribute names and associated value descriptions matching class attribute names and associated values possessed by a class included in the virtual-machine-executable software; the set including an attribute name of a set of attribute names including

i. a class name,

ii. an interface name,

iii. a parent class name,

iv. an inherited method name,

v. a defined method name,

vi. a private method name,

vii. an inherited field name,

viii. a defined field name,

ix. a private field name,

x. a constant value attribute,

xi. a synthetic attribute,

xii. a code attribute,

xiii. an exception attribute, and

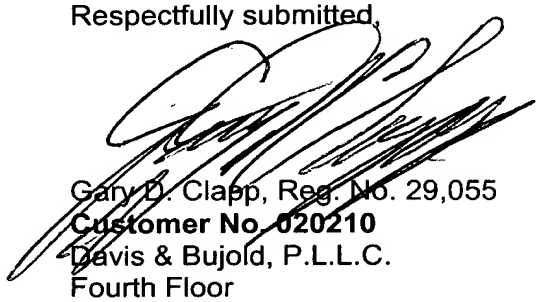
xiv. a depreciated attribute"; or of

"(b) searching for a class possessing a class attribute name and an associated value consistent with the set of specified class attribute name and associated value description;" or of

"(c) classifying the class possessing a attribute name and associated value consistent with the set of specified class attribute names and associated value descriptions as the target class" to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. § 103.

In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,



Gary D. Clapp, Reg. No. 29,055
Customer No. 020210
Davis & Bujold, P.L.L.C.
Fourth Floor
500 North Commercial Street
Manchester NH 03101-1151
Telephone 603-624-9220
Facsimile 603-624-9229
E-mail: patent@davisandbujold.com

APPENDIX

PENDING CLAIMS

1. A method for instrumenting object oriented virtual-machine-executable software comprised of class files containing bytecode, each class file being defined by a class associated with a class-hierarchy location and having a corresponding class name, the method comprising the steps of:

(a) identifying a class included in the software as a target class wherein the target class is associated with a first class-hierarchy location and with a first class name;

(b) instrumenting the bytecode of the target class by one of

(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form; and

(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, the bytecode of the target class remaining in the original, unmodified form; and

(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new

instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form; and,

(c) causing a virtual machine to process as the target class the new instrumented class assigned the first class name.

2. The method of claim 1 including the following by the steps of:

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the target class;

(ca) after completing step (c), un-suspending operation of the virtual machine.

3. The method of claim 1 including the following by the steps of:

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the target class to execute a software program that executes step (c);

(ca) upon termination of the software program that executes step (c) resuming the operation of the virtual machine.

4. The method of claim 1 wherein the virtual-machine is a Java virtual machine and the step of identifying the target class included within the virtual machine executable software includes the steps of:

(a) specifying a set of class attribute names and associated value descriptions matching class attribute names and associated values possessed by a class included in the virtual-machine-executable software; the set including an attribute name of a set of attribute names including

xv. a class name,

xvi. an interface name,

xvii. a parent class name,

xviii. an inherited method name,

xix. a defined method name,

xx. a private method name,

xxi. an inherited field name,

xxii. a defined field name,

xxiii. a private field name,

xxiv. a constant value attribute,

xxv. a synthetic attribute,

xxvi. a code attribute,

xxvii. an exception attribute, and

xxviii. a depreciated attribute,

(b) searching for a class possessing a class attribute name and an associated value consistent with the set of specified class attribute name and associated value description; and

(c) classifying the class possessing a attribute name and associated value consistent with the set of specified class attribute names and associated value descriptions as [[a]] the target class.

5. The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using an interface provided by the virtual machine.

6. The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using a virtual machine that is modified to perform this capability.

7. The method of claim 1 wherein the virtual machine is a JAVA™ virtual machine.

8. The method of claim 1 wherein the virtual-machine is a JAVA™ virtual machine and the steps of adding instrumentation to the target class include adding, removing, modifying, reordering or substituting a named class component of a set of named class components including:

(i) a class name,

(ii) a super class name,

(iii) an interface index array,

(iv) a field table,

(v) a method table,

(vi) a constant pool,

(vii) an attribute table,

(viii) an index array, and

(ix) access flags.

9. The method of claim 6 wherein the class component is represented by a plurality of memory locations, the step of modifying a class component including adding, removing, reordering or substituting a memory locations representing the class component or changing the content of a memory locations representing the class component.

10. The method of claim 1 wherein the target class is represented by a plurality of memory locations, the step of adding instrumentation to the target class including adding, removing, or reordering or substituting a memory location representing the target class or changing the contents of a memory l location representing the target class.

11. The method of claim 1 wherein the step of assigning the new instrumented class to a class hierarchy location adjacent to and above the class hierarchy location associated with the target class includes:

modifying the new instrumented class to recognize a super class associated with the target class as the super class associated with the new instrumented class;

modifying the target class to recognize the new instrumented class as the super class associated with the target class.

12. The method of claim 1 wherein the step of assigning the new instrumented class to a class hierarchy location adjacent to and below the class hierarchy location associated with the target class includes modifying the new instrumented class to recognize the target class as the super class associated with the new instrumented class.

13. An apparatus for instrumenting object oriented virtual-machine-executable software comprised of class files containing bytecode, each class file being defined by a class-associated with a class-hierarchy location and having a corresponding class name, comprising:

(a) a target class identifier that identifies a class included in the software as a target class wherein the target class is associated with a first class-hierarchy location and with a first class name;

(b) an instrumenter instrumenting the bytecode of the target class by one of

(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, the bytecode of the target class remaining in an original, unmodified form;

(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a

second class name to the target class, the bytecode of the target class remaining in the original, unmodified form; and

(iii) creating a new instrumented class separate from the target class and including instrumentation, assigning the first class name to the new instrumented class and assigning a second class name to the target class whereby the new instrumented class is recognized as the target class and the target class is inactive with the bytecode of the target class remaining in the original, unmodified form; and

(c) a virtual machine for processing as the target class the new instrumented class assigned the first class name.

14. The apparatus for instrumenting object oriented virtual-machine-executable software of claim 13, further including class identification apparatus for identifying a class included within object oriented virtual-machine-executable software, the class identification apparatus comprising:

a class query engine which receives as input a set of class attribute names and associated value descriptions; and

a class searcher that sequentially searches classes included in the software and performs a predetermined action upon identifying a class that possesses a set of class attribute names and associated values that match the set of class attribute names and associated value descriptions as the target class.